

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A computer implemented distributed classification system comprising:

a processor;

a memory communicatively coupled to the processor, the memory having stored therein computer executable instructions to implement the system, including:

a plurality of software components shared by unrelated software design tools, ~~stored in a computer readable storage medium and executable by a processing device~~, wherein each of the design tools controls at least one of the software components; and

a classification component that couples the software components to a common classification structure based on a structure type comprising structure type class that represents a name of the structure type and identifies if the common classification structure is a hierarchy, node types defining at least one type of software component supported by the common classification structure and structural constraints, the structural constraints define the permissible parent-child relationships between the various node types ~~and wherein a plurality of applications access the software components;~~ and

an event monitoring system that monitors the common classification structure and automatically provides an opportunity to prevent changes to the common classification structure by communicating at least one notification prior to implementing the changes to the common classification structure,

wherein the memory stores the classification component and the coupling between the software components and the common classification structure.

2. (Cancelled)

3. (Previously Presented) The system of claim 1, wherein the classification structure is hierarchical.
4. (Original) The system of claim 1, wherein the software components are associated with classification nodes.
5. (Original) The system of claim 4, wherein a classification node comprises a globally unique identifier.
6. (Currently Amended) The system of claim 4, ~~wherein further comprising~~ a graphical user interface ~~is employed~~ employable by a user to ~~classify~~ associate a classification node to at least one of the software components.
7. (Currently Amended) The system of claim 6, wherein the graphical user interface enables a user to drag[[s]] and drop[[s]] a displayed representation of the at least one software component[[s]] onto a displayed representation of the classification node to assign the at least one software component to the classification node.
8. (Original) The system of claim 1, wherein the classification component utilizes heuristics and statistical analysis related to artificial intelligence to couple software components to the common structure.
9. (Cancelled)
10. (Currently Amended) The system of claim 1, further comprising a notification component that alerts consumers of the common structure ~~of~~ upon committing a change to the common structure.

11. (Currently Amended) A computer implemented software tool interaction system comprising:

a processor;

a memory communicatively coupled to the processor, the memory having stored therein computer executable instructions to implement the system, including:

means for generating a common classification scheme amongst a plurality of unrelated software tools stored in a computer readable storage medium of the memory, wherein the classification of ~~the~~ a plurality of components controlled by the plurality of design tools is based on a structure type and comprises structure type class that describes how the plurality of components are arranged into a hierarchy associated with the classification scheme, node types that define types of components that can be included in the hierarchy and structural constraints~~[[,]]~~ ~~the structural constraints~~ that define the permissible parent-child relationships between the various node types; ~~and~~

means for maintaining the common classification scheme to provide a foundation for a cohesive user experience and wherein the plurality of unrelated software design tools access the components; and

means for monitoring the plurality of components that automatically communicates notifications to users prior to implementing modifications to the hierarchy such that users have an opportunity to prevent undesirable changes wherein further notifications are provided to the users upon implementing the modifications to the hierarchy.

wherein the memory stores the classification component and classification of the components.

12. (Currently Amended) The system of claim 11, wherein the means for generating includes a user generates the classification scheme employing a graphical user interface allowing a user to drag and drop artifacts graphically displayed components onto graphically displayed classification nodes of a displayed structure.

13. (Currently Amended) The system of claim 11, wherein the means for generating a classification scheme is generated includes means for automatically utilizing heuristics and

statistical analysis associated with artificial intelligence to automatically define the common classification scheme.

14. (Cancelled)

15. (Cancelled)

16. (Currently Amended) A computer implemented common classification methodology, comprising:

employing a processor to execute computer executable instructions stored in memory to perform the following acts:

receiving user input related to at least one taxonomy based on a structure type comprising a structure type class that holds name of the structure type and identifies if the common classification structure associated with the taxonomy is a hierarchy, node types defining type of artifacts that are to be included in the taxonomy and structural constraints defining permissible parent-child relationship between various node types;

instantiating a common classification structure for the taxonomy based at least on the structure type;

generating one or more taxonomies comprising defining node types, structure type classes and structural constraints, wherein the parent-child relationship between the various node types is specified by the structural constraints;

exposing the common classification structure among a plurality of unrelated software design tools as one or more typed XML (eXtensible Markup Language) documents wherein the nodes are typed according to the structure type;

maintaining the taxonomies to facilitate interaction with taxonomy artifacts by [[a]] the plurality of unrelated software design tools, wherein each of the design tools controls at least one of the taxonomy artifacts[[.]];

monitoring the common classification structure to detect manipulations of the structure;

providing automatic notifications to users upon receiving input manipulating the common classification structure;

receiving user feedback in response to the notifications;

allowing the manipulation of the common classification structure based on the user feedback; and

informing the users of the manipulations to the common classification structure.

17-18. (Cancelled).

19. (Original) The method of claim 16, wherein each node in a taxonomy is identified by an immutable globally unique node identifier.

20. (Currently Amended) The method of claim 16, further providing a graphical user interface to the users for generating the taxonomy wherein the taxonomy is generated by a user employing a graphical user interface, wherein the taxonomy is generated by a user employing a graphical user interface.

21. (Currently Amended) The method of claim 20, ~~wherein a user generates the taxonomy by dragging and dropping artifacts onto a classification node,~~ further comprising facilitating association of the taxonomy artifacts to respective node types by receiving an indication of a drag and drop of at least one of the taxonomy artifacts into respective node type.

22. (Currently Amended) The method of claim 16, further comprising generating ~~wherein the taxonomy is generated~~ automatically by ~~a component~~ employing heuristics and statistical analyses related to artificial intelligence.

23. (Cancelled)

24. (Cancelled)

25. (Cancelled)

26. (Cancelled)

27. (Cancelled)

28. (Currently Amended) A common enterprise classification scheme methodology comprising:

employing a processor to execute computer executable instructions stored in memory to perform the following acts:

receiving input related to classifying a plurality of artifacts controlled by a plurality of unrelated software tools in accordance with a taxonomy scheme;

instantiating a common structure based on a structure type associated with the received input, the common structure comprising structure type class that details the arrangement of taxonomy artifacts into a hierarchy associated with the common structure, node types which specify types of artifacts that can be included in the common structure and structural constraints~~[[.]]~~ which the structural constraints define the permissible parent-child relationship between the various node types;

exposing the common structure amongst a plurality of unrelated software design tools to facilitate the classification of the artifacts; classify the components they control, and provide a foundation for a cohesive user experience, wherein the plurality of tools access the components~~[[.]]~~;

detecting one or more changes to the common structure;

automatically notifying one or more of users or owners of the artifacts regarding changes to the common structure;

receiving feedback from the owners or users regarding the notifications;

preventing changes to the common structure if the users or owners veto the changes in the feedback;

implementing the changes if no veto is received; and

informing the owners and users of the changes implemented to the common structure.

29. (Cancelled).

30. (Original) The method of claim 28, wherein the common structure is exposed *via* a graphical user interface.

31. (Cancelled)

32. (Cancelled)

33. (New) The method of claim 20, further comprising receiving user input regarding a node within the common classification structure via the graphical user interface.

34. (New) The system of claim 1, the common classification structure is provided to the software design tools as one or more XML documents, wherein nodes comprised in the XML documents are typed in accordance with respective structure types.